UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/458,139 | 12/09/1999 | YI-MIN WANG | 200730 | 8486 |

7590 05/22/2003

VLADAN M VASIJEVIC
LEYDIG VOIT & MAYER LTD
TWO PRUDENTIAL PLAZA SUITE 4900
180 NORTH STETSON
CHICAGO, IL 60601

| EXAMINER | |
|---|---|
| ZHEN, LI B | |
| **ART UNIT** | **PAPER NUMBER** |
| 2126 | 8 |

DATE MAILED: 05/22/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PRL

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/458,139 | WANG ET AL. | |
| | Examiner | Art Unit | |
| | Li B. Zhen | 2126 | |

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence addr ss* --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *06 March 2003* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-46* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-46* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. §§ 119 and 120**

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

## DETAILED ACTION

### *Claim Rejections - 35 USC § 112*

1. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claim 12 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

3. Claim 12 recites the limitation "the send buffer" in line 1. There is insufficient antecedent basis for this limitation in the claim.

### *Claim Rejections - 35 USC § 103*

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1, 2, 4, 5, 14 – 18, 24, 25, 27, 28, and 37 – 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Harnessing User-Level Networking Architectures for Distributed Object Computing over High-Speed Networks" (hereinafter Madukkarumukumana) in view of "Virtual Interface Architecture Specification, Revision 1.0" (hereinafter VIA) and "COMERA: COM Extensible Remoting Architecture" (hereinafter COMERA).

As to claim 1, Madukkarumukumana teaches (p. 5, Section 4.2 Anatomy of Custom Stub/Proxy; p. 2, Section 2. Virtual Interface Architecture) a method of communication (p. 4, Section 4. DCOM Remote Method Invocation over VI Architecture Transport) between a first object (custom proxy) on a first computer (client process/machine) and a second object (custom stub) on a second computer (server process/machine), RPC buffer (receive and reply buffers) the computers connected by a network (VI Architecture is a user-level networking architecture, Section 2. Virtual Interface Architecture, p. 2), and calling an interface of the second object with the first object (user-level VI transport for inter-process communications, Fig. 5). Madukkarumukumana does not describe the transmission process in detail.

However, VIA teaches (p. 12 – 13, Section 2.1.1. Virtual Interfaces) placing in the buffer (send queue) a copy of a first pointer (Descriptor is a data structure that contains all of the information that the VI Provider needs to process the request, such as pointers to the data buffers) to a first parameter (data stored in the data buffers), the network interface card transmitting the first parameter pointed to by the first pointer by reading the first parameter out of the first memory location (VI NIC directly performs data transfer functions), and treating the first pointer as a scatter-gather entry (p. 30, Section 6.1.1.1. Scatter-Gather Considerations).

It would have been obvious to apply placing in the buffer a copy of a first pointer to a first parameter, transmitting the first parameter by the network interface card as taught by VIA to the invention of Madukkarumukumana because it would avoid intermediate copies of the data and bypasses operating system to achieve low latency,

high bandwidth data transfer (p. 2, Section 2. Virtual Interface Architecture of Madukkarumukumana).

Although Madukkarumukumana (p. 2, left column, lines 1 – 7) teaches replacing legacy RPC transports in DCOM, Madukkarumukumana also teaches (p. 4, Section 4) the use of custom marshalling. Custom marshalling allows custom remoting architecture to replace standard remoting architecture. Madukkarumukumana does not describe custom marshalling in detail.

However, COMERA (section 3.2. The COMERA architecture) teaches COM extensible architecture that uses custom marshalling to rebuild the standard remoting architecture, which includes a COMERA RPC channel object. Although custom marshalling allows an object to bypass the standard remoting architecture, it also constructs a custom remoting architecture, which would also include RPC mechanisms such as RPC channel object and a RPC layer.

It would have been obvious to apply the teaching of using a custom marshalling to construct a custom remoting architecture which includes RPC mechanisms such as RPC channel object and a RPC layer as taught by COMERA to the invention of Madukkarumukumana as modified because custom marshalling provides the basis for extensibility in COM remoting architecture (section 3.1 of COMERA).

As to claim 24, this is a product claim that corresponds to method claim 1; note the rejection of claim 1 above, which also meets this product claim.

As to claims 2 and 25, Madukkarumukumana as modified teaches (p. 15, first paragraph of VIA) issuing a notification on the first computer after the network interface

card has finished reading the first parameter out of the first memory location (the Send/Receive model of data transfer requires that the VI Consumers be notified of Descriptor completion at both ends of the transfer).

As to claims 4 and 27, these claims are drawn to placing one or more pointers in the buffer and the network interface card transmitting the parameters that the pointers point to. Madukkarumukumana as modified teaches the buffer (send queue, p. 10 of VIA) contains one or more pointers (Descriptors that describe the data to be transmitted, p. 10 of VIA) and asynchronously processing the posted Descriptors (p. 13 of VIA). As to RPC buffer and scatter-gather entry, see the rejection to claim 1 above.

As to claims 5 and 28, these claims are drawn to issuing a notification on the sending computer each time the network interface card has finished reading a parameter. Madukkarumukumana as modified teaches (p. 15, first paragraph of VIA) the Send/Receive model of data transfer requires that the VI Consumers be notified of Descriptor completion at both ends of the transfer.

As to claim 14, Madukkarumukumana teaches (p. 5, Section 4.2 Anatomy of Custom Stub/Proxy; p. 2, Section 2. Virtual Interface Architecture) a method of communication between (p. 4, Section 4. DCOM Remote Method Invocation over VI Architecture Transport) a first object (custom proxy) located on a first computer (client process/machine) and a second object (custom stub) located on a second computer (server process/machine), the first and second computers connected by a network (VI Architecture is a user-level networking architecture, Section 2. Virtual Interface Architecture, p. 2), and receiving a call from the first object on an interface of the second

object (user-level VI transport for inter-process communications, Fig. 5), and accessing the parameter by the second object (interface stub unmarshals method parameters from receive buffers and dispatches actual object methods). As to RPC layer and RPC buffer, see the rejection to claim 1 above. Madukkarumukumana does not describe the receiving process in detail.

However, VIA teaches (p. 12 – 13, Section 2.1.1. Virtual Interfaces) receiving by the network card a parameter of the call from the first object (VI NIC directly performs data transfer functions), and storing the parameter in a memory location (receive queue contains Descriptors that describe where to place incoming data, p. 10).

It would have been obvious to apply receiving by the network card a parameter of the call from the first object, and storing the parameter in a memory location as taught by VIA to the invention of Madukkarumukumana because it would bypass operating system to achieve low latency, high bandwidth data transfer (p. 2, Section 2. Virtual Interface Architecture of Madukkarumukumana).

As to claim 37, this is a product claim that corresponds to method claim 14; note the rejection of claim 14 above, which also meets this product claim.

As to claims 15, 16, 38, and 39, Madukkarumukumana teaches that the memory location is the RPC buffer and accessing the parameter is performed in the RPC buffer (each interface stub unmarshals method parameters from receive buffers, p. 5, Section 4.2 of Madukkarumukumana).

As to claims 17, 18, 40, and 41, Madukkarumukumana as modified teaches the memory location is the memory storage location (physical memory) and accessing the

parameter in the memory storage location (locking the pages of a virtually contiguous memory region into physical memory, Section 2.2, p. 14 of VIA).

6. Claims 3, 6, 7, 26, 29 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Madukkarumukumana, VIA, and COMERA in view of U.S. Patent No. 6,131,126 to Kougiouris.

As to claims 3, 6, 7, 26, 29, 30, Madukkarumukumana as modified teaches reclaiming memory (reuse registered memory buffers, Section 2.2, p. 14 of VIA) but does not specify reclaiming the memory location after receiving the notification.

However, Kougiouris teaches (column 2, lines 28 – 45) a method in a computer system for inter-process communication that reclaims a memory location after data transmission (the first buffer is deallocated upon receipt of the communication).

It would have been obvious to apply the teaching of reclaiming a memory location after data transmission as taught by Kougiouris to the invention of Madukkarumukumana as modified because this prevents large and unnecessary consumption of memory resources.

7. Claims 8, 13, 19, 20, 31, 36, 42, and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Madukkarumukumana, VIA, and COMERA further in view of U.S. Patent No. 6,044,409 to Lim.

As to claims 8, 13, 31, and 36, Madukkarumukumana as modified teaches a first send buffer, a first receive buffer (VI Consumer at the receiving end pre-posts a Descriptor to the receive queue, first paragraph, p. 15 of VIA), and the first receive buffer is posted to be of sufficient size to accept the second data (VI Consumer on the

receiving side must post a Receive Descriptor of sufficient size before the sender's data arrives, second full paragraph, p. 15 of VIA). Madukkarumukumana as modified teaches posting a receive buffer before the data arrives but does not specify posting on the first computer a first receive buffer prior to sending a first data to the second computer.

However, Lim teaches (column 12, lines 19 – 25, 55 – 60, and 64 – 66) posting on the first computer a first receive buffer prior to sending a first data to the second computer (a marshal buffer appropriate for the transport selected is created in step 206, Fig. 4), the first receive buffer will receive a second data from the second computer in response to the first data (the client receives a reply from the server and encapsulates the reply in a marshal buffer 216 and 218, Fig. 4), and sending the first data to the second computer (the contents of the marshal buffer are transmitted over the selected transport to the identified end point 212, Fig. 4).

It would have been obvious to apply the teaching of posting on the first computer a first receive buffer prior to sending a first data to the second computer as taught by Lim to the invention of Madukkarumukumana as modified because this would ensure that there is memory available to store the response data.

As to claims 19 and 42, Madukkarumukumana as modified teaches storing on the second computer a second data into a first receive buffer (marshals return parameters into the reply buffers, Section 4.2, p. 5 of Madukkarumukumana). As to posting a receive buffer prior to sending data to a computer and the first receive buffer

was posted to be of sufficient size to accept the second data, see the rejection to claims 8 and 13 above.

As to claims 20 and 43, Madukkarumukumana as modified teaches (column 12, lines 55 – 67 of Lim) the first data from a send buffer to the first computer was sent (transmit contents of marshal buffer over selected transport to identified end point 212, Fig. 4) prior to receiving the second data form the first computer (receive reply from server 216, Fig. 4).

8. Claims 9 – 12 and 21 – 23, 32 – 35, and 44 – 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Madukkarumukumana, VIA, COMERA and Lim further in view of Kougiouris.

As to claims 9, 11, 21, 23, 32, 34, 44, and 46, these claims are drawn to cleaning up a buffer on a computer after the data from the buffer has been transmitted. Note the rejection of claims 3, 6, 7, 26, 29, and 30 above.

As to claims 10 and 33, these claims are drawn to posting a receive buffer prior to data transmission. See the rejection to claims 8, 13, 31, and 36 above.

As to claims 12, 22, 35, and 45, Madukkarumukumana as modified teaches using a send buffer to send data to a computer (transfer data directly between buffers of a VI Consumer and the network, Section 2.2, p. 14 of VIA).

***Response to Arguments***

9. Applicant's argument that Madukkarumukumana does not teach a RPC layer because Madukkarumukumana teaches custom marshalling which bypasses all of the RPC mechanisms (p. 10, lines 4 – 12). Madukkarumukumana replaces legacy RPC

transports using custom marshalling. COMERA teaches that custom marshalling can be used to create custom remoting architecture. COMERA (section 3.2. The COMERA architecture) teaches COM extensible architecture that uses custom marshalling to rebuild the standard remoting architecture, which includes a COMERA RPC channel object. Although custom marshalling allows an object to bypass the standard remoting architecture, it also constructs a custom remoting architecture, which includes RPC mechanisms such as RPC channel object. Custom marshalling in Madukkarumukumana would also provide the DCOM objects with a custom remoting architecture that would include RPC mechanisms because DCOM is an extension of COM. Therefore, the standard RPC transports in Madukkarumukumana are replaced with a custom RPC layer through the use of custom marshalling.

The applicant argues, "Kougiouris describes the allocation of memory space in an operating system on which both communication applications are running...however, the communicating programs described by the present application do not share a common operating system..." (p. 11, lines 2 – 9). The examiner respectfully disagrees because Kougiouris clearly teaches the communication applications can be running on the same computer or separate computers (the remote process may be resident either in a local processor or computer system's memory or on a remote processor in a remote computer system in a distributed environment).

## *Conclusion*

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (703) 305-3406. The examiner can normally be reached on Mon - Fri, 8am - 4:30pm.

The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Li B. Zhen
Examiner
Art Unit 2126

lbz
May 16, 2003